

AAASeed training

Train2

Us

- Mâa father of **AAASeed**
- Louis Montagne
 - helping me with the **AAAFoundation**
 - in charge also of the installer/updater
 - that we will not use today !
 - Chat Gpt tool
 - feed with **AAASeed** Corpus
- Abdallah Meskine
 - AAAPadawan**
 - direct your first level questions to him
 - abdallah.meskine@gmail.com

You Train 1

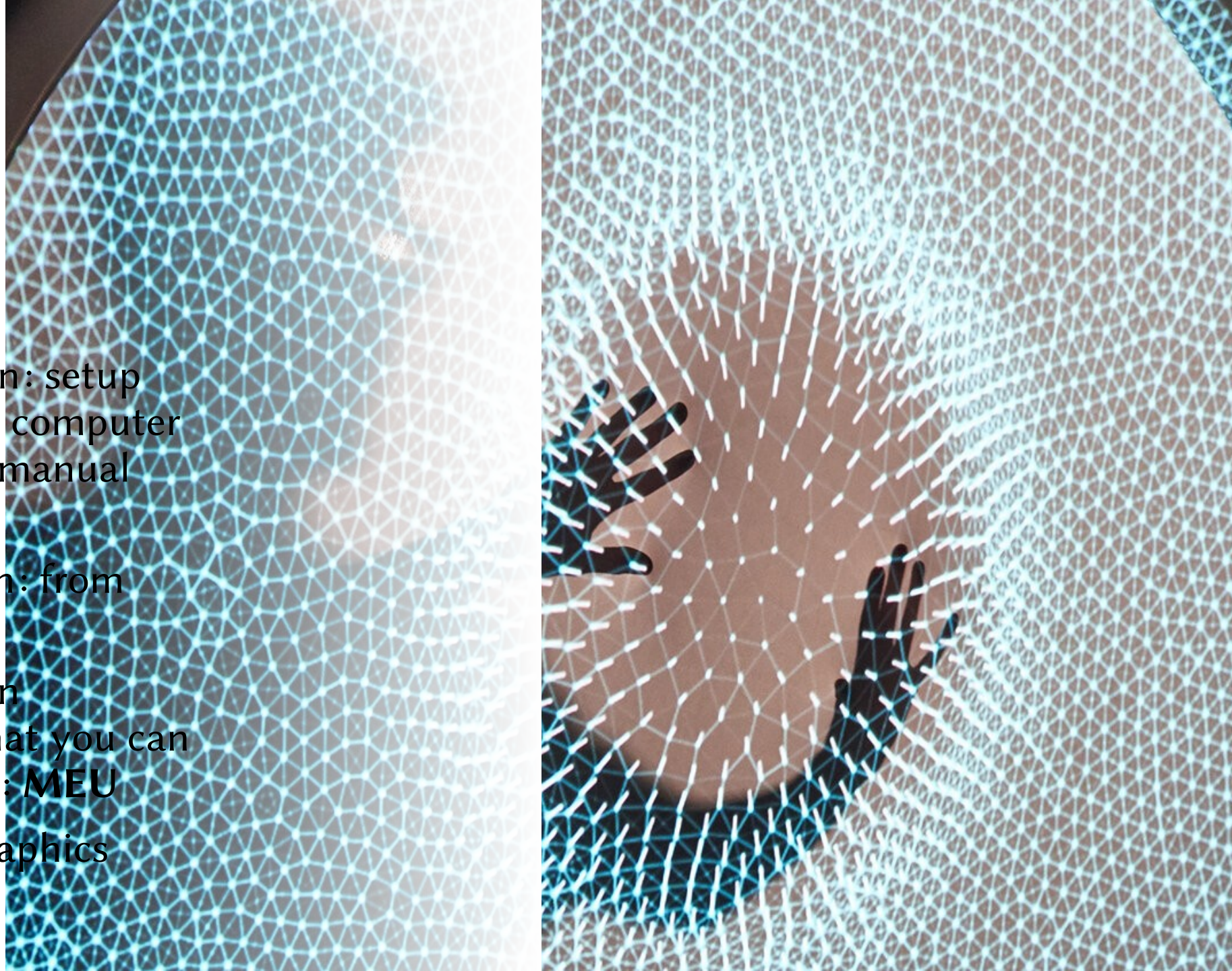
- Jaime
jaimitocdm@free.fr
- Tendayi
tendayi.vine@gmail.com
- Tom Dale
tom@tomdale.org.uk
- Christopher Thomas Allen
chris@thelightsurgeons.co.uk
- Joe Strickland
chronicinsanitytheatre@gmail.com
- Pawel Janicki
pawel@wrocenter.pl
- Giannis Dimitriadis
johndim@iti.gr

You Train 2

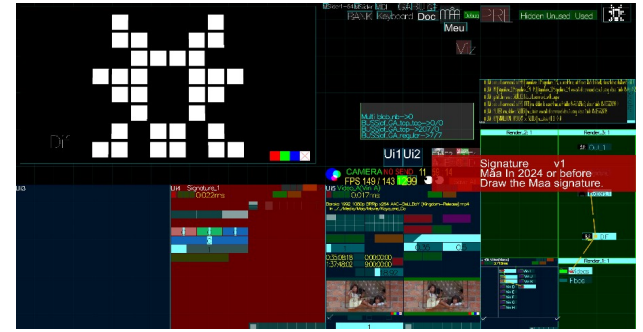
DAY-1

1PM to 7PM

- Onboarding session: setup **AAASeed** on your computer from scratch : the manual way
- Basic Manipulation: from **BU to MEU**
- Generic use & main concepts: learn what you can do with **AAASeed: MEU**
- Some computer graphics
- Architecture

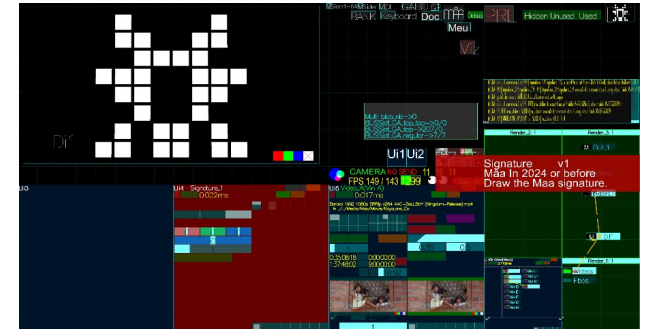


What is AAASeed ?



- A generic toolkit to build realtime process
- oriented mainly in 3 contexts
 - Video Jockey
 - Open data
 - interactive Art installation
- Soon OpenSource (MIT license) and free
- Brainchild of Maa
 - Used and refined in production for 25 years
 - Cleaned and stabilized during the ArtCast4d EU Project

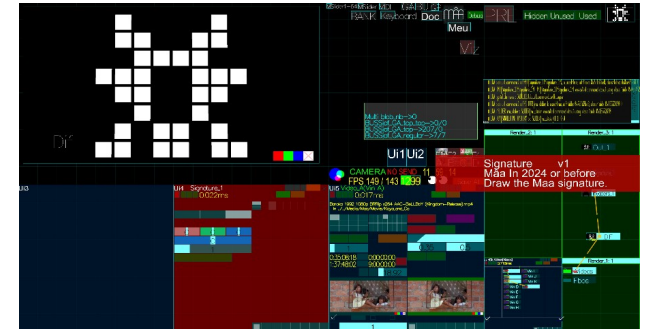
For whom is **AAASeed** ? Artist



- A place where I can assemble and compose blocks on the fly
- most of these block deal with graphics process
- some of the blocks
 - deal with inputs (cameras, captors)
 - analyse images
 - connect elements
- so I can produce live graphic and or interactive processes

For whom is **AAASeed** ?

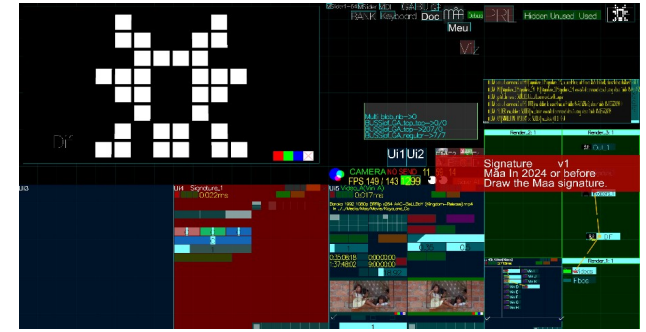
Developer



- On the base of an old school robust C++ rendering graph using the very fast and flexible Lua scripting language you can access the low level fonctionnality:
Window system, custom C object, OpenGL, C++ objects, Gl Shaders, OpenCl, OpenCV, Nvidia Flex, dlib, bullet,
...
- A rich lua virtual machine provide
an interface to edit and customize realtime processes
most of it happen live while processes are runing
you can extend and edit on the fly the interface

User point of view

- This a PC windows application for now
AAASeed_Metal.exe
- tested under windows 10 and 11
should also work on Windows 8,7 even XP.
- It can run on low end machines
even an executable for non Avx2 Processor (AAASeed_Wood.exe)
- But it likes fast machines and use the Gpu a lot
Graphic Processor Unit
Love NVidia but function on integrated Intel
support Amd most of the time (getting better on this every month)



Shadoks point of view

- talking with the **GA**

Global Action: the top level

- using **BU**

Box User: element of interface

BU regrouped in **BUS**

BUS contain **BU** which can contain one

BUS which can contain **BUS** which can contain one

– **BUS** which can contain **BUS** which can contain one

- you manipulate a bunch of **MEU**

Module Editable Unit: fonctionnal editable blocks

sometime represented in their short form the **MU**: Module Unit

- We lost the **ZO** but we will find it, promised
- More on this: search shadoks on YouTube
 - <https://www.youtube.com/watch?v=SlA57Zw-FN4>
- Thanks to **aaa**production



You better pump even if nothing happen than to risk than something worse happen while not pumping.



Install 1/5

- Create Folders

- AAA** : where your **AAASeed** stuff and work will be

- Media** where you put content as videos, images, sounds...
 with sub folder by projects

- Install** where you put associated installers, drivers...

- Dll** where we will store Dlls associated with **AAASeed**

Install 2/5

- <https://aaaseed.org/training/>
- <https://192.168.1.42:5001/webman/index.cgi>

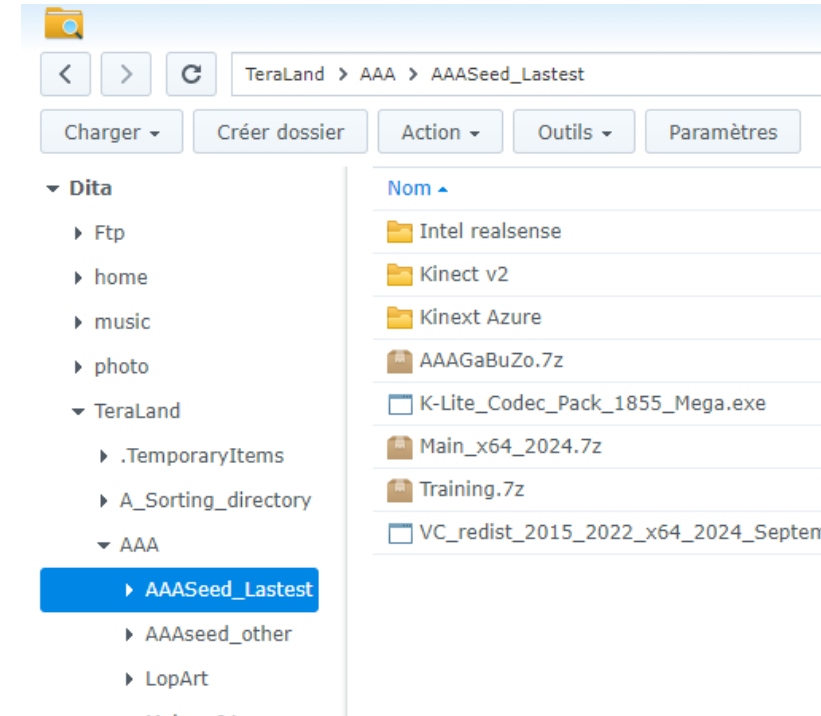
yes trust it

- look it
- FileStation



Goto Dita / Teraland / AAA / AAASeed_Lasttest

- Download
 - AAAGaBUZo.7z : AAASeed with environements
 - Main_x64_2024.7z : Dll associated
 - VC_redist_2015_2022exe and vcredist_2013_x64.exe
needed installers for bricks used by AAASeed
 - Training.7z : Medias examples for the training



Install 3/5

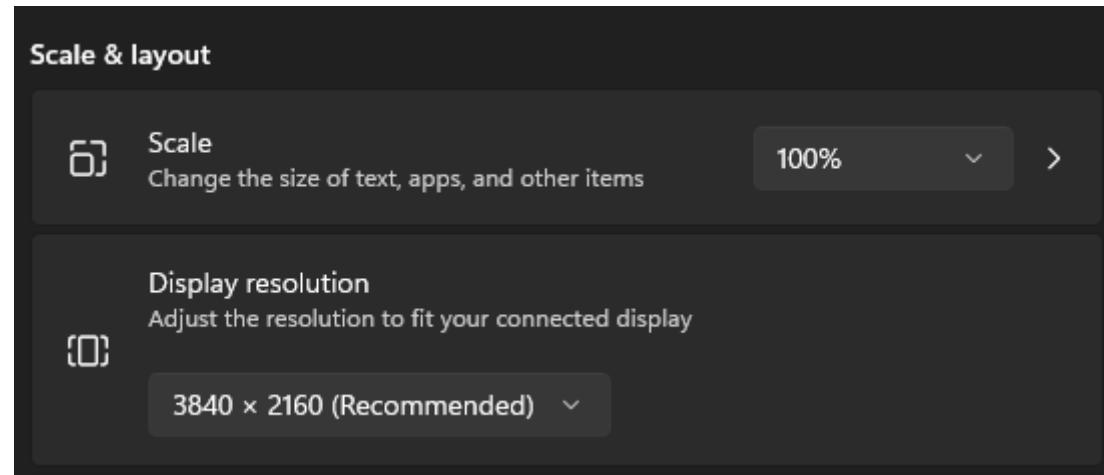
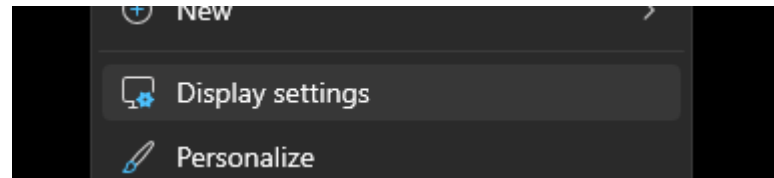
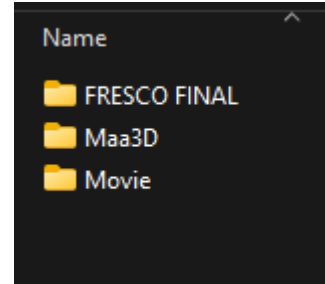
- https://www.codecguide.com/download_kl.htm
klite Mega
handle decompression of video
- <https://www.7-zip.org/>
Windows 64-bit x64 version
handle compression and decompression of 7z files
- <https://code.visualstudio.com/>
Download for windows
text editor
- <https://www.xnview.com/en/>
MP or Classic
Image browser with batch and conversion capabilities

Install 4/5

- 7z ... -x64.exe : execute to install
- decompress AAAGaBUZo.7z in AAA
- launch AAASeed_Metal.exe in it
 - Error because AAASeed need some stuff from the system
 - Place it in the TaskBar
- Install system stuff
 - decompress AAAGaBUZo.7z in AAA/Dll and check inside
 - Copy Dll path
 - This PC / Properties / Advanced system settings / Environement Variables
 - User variable / Edit... / New
 - Paste path and eventually change order then validate
 - Execute VC_redist_2015_2022exe and eventually vc_redist_2013_x64.exe
- Launch AAASeed_Metal.exe
 - should work
- double esc to save and exit

Install 5/5

- Decompress Training.7z in AAA/Media
media examples for our training
- Install
 KLite codec
 default options all the way
 VSCode
 XnView optional
- Configure graphics output
 right click on desktop / Display settings
 resolution
 position
 set scale at 100% on every screen



AAASeed Sources

- Mâa's NAS we used
- Installer / Updater done by Louis Montagne
will go back to it
- but wanted you to be free and have no dependency on anything
put all the install Folder on a USB key (except VS Code)

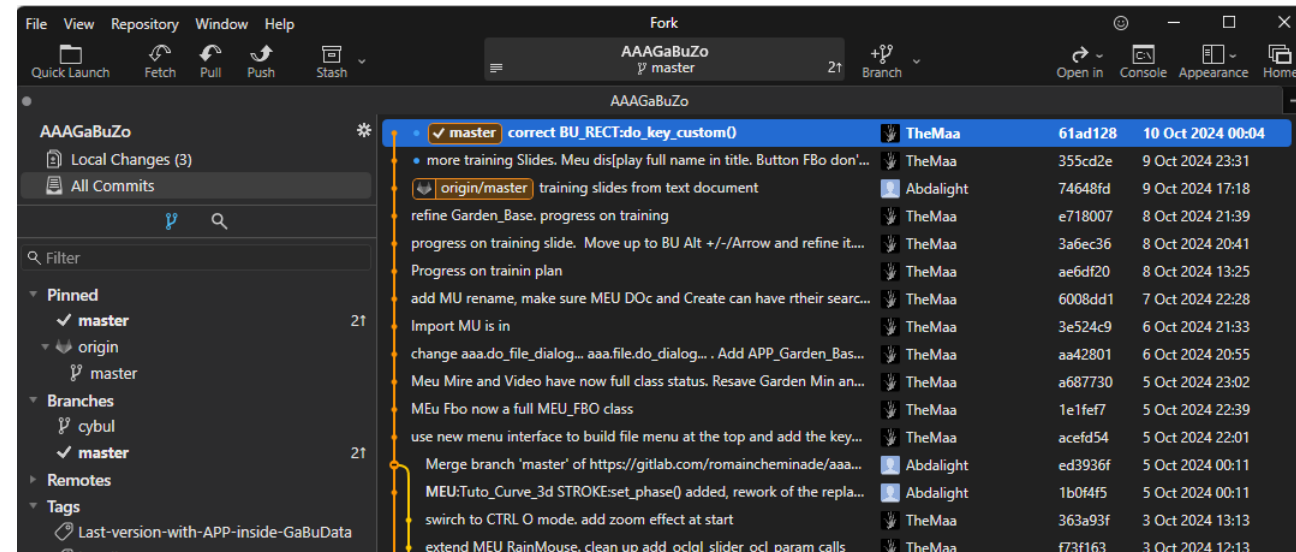
- Developer access
git

-AAAGaBuZo
-Apps
-Lua

Svn

-**AAASeed** exe and dll
-C++
-will move to git at some point

request Mâa if you want it

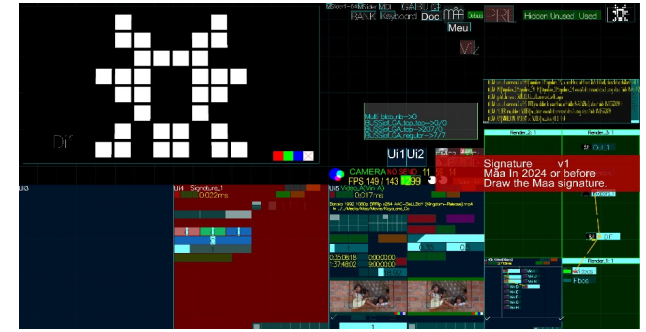


AAASeed first look at file system

- **AAAGaBuZo** Folder
- **AAADoc**
 - **Lua** API
 - lua_aaaseed_draw.lua
 - lua_aaaseed_interface.lua
 - **Training**
 - Slide and Pdf
- **AAAKernel**
 - **Developer** space
 - Some examples
 - Fonts
 - Textures...
- **AAAUser**
 - **AAA_Who.txt**
- **APP**
 - **APPlication** examples

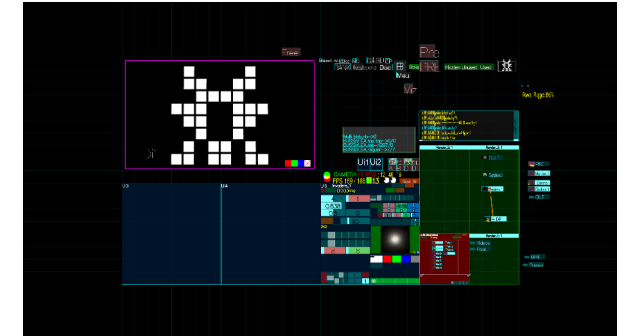
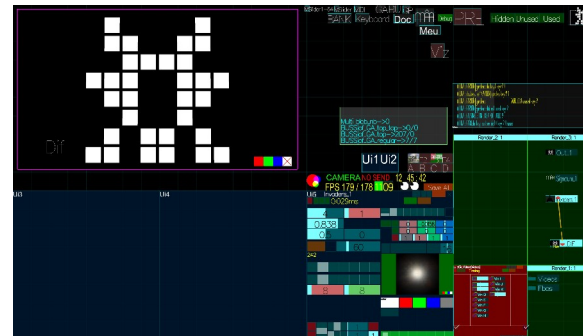
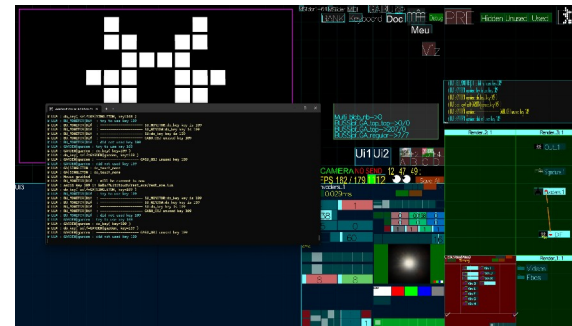
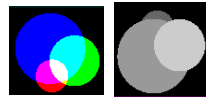
AAASeed Basic

- keyboard first step
- **w** like (w)indow
switch between window and full screen Module
- **Tab**
switch Flatland on and Off
Ctrl e like (e)dit
switch edit mode (more later on this)
see at the top of Flatland
Red LOCK vs Green Edit
leave it as Edit
- **Ctrl w** and **Shift Ctrl w**
go through all the possibility full screen on different Screen
Shift go the other way
- **Double Esc** to Quit (save global stuff)



AAASeed Basic

- Don't be scared
need to get used but efficient
- Careful with shift lock
Dangerous
- **Ctrl F2**
Force the GaBuZoMeu interface On and move it from screen to screen
- **F2**
Only flip the GaBuZoMeu interface
- **F3**
Switch On/Off the rendering
BU_Alive
- **F1** or **Ctrl h** like (h)elp
switch the help system
- **m** like (m)essage
pop the terminal window
- **Ctrl Tab**
Change the UI Scale



BU Box User

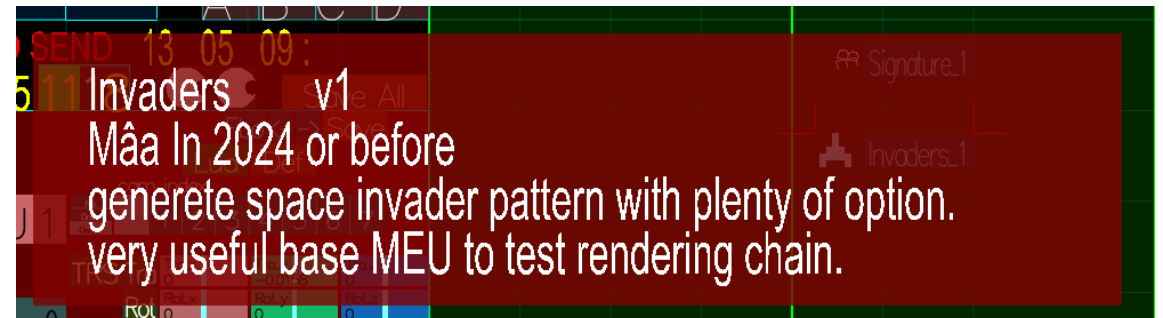
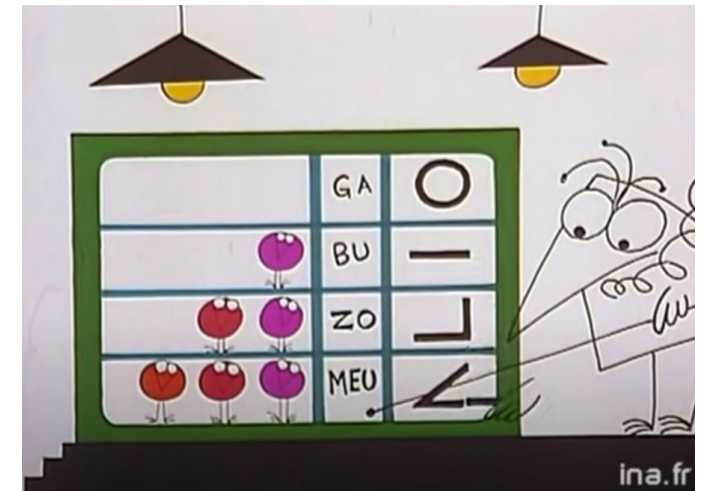
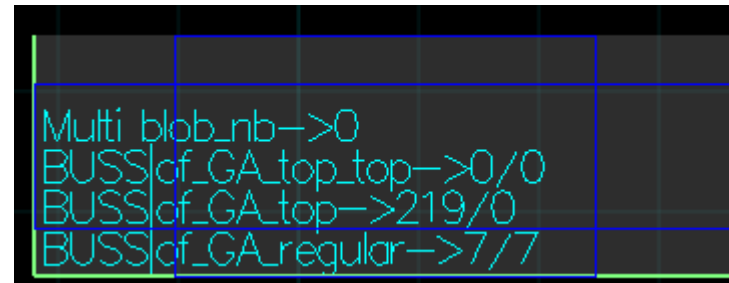
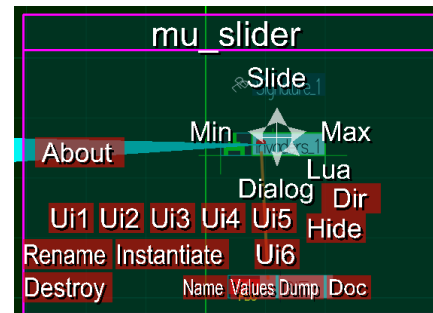
- Every UI element is a **BU**
- Exploration from bottom
- **BU** can contain **BUS**
BUS / BUS / BUS / BUS

- Click
 - simple with left button
 - double
 - triple
 - long

- **Alt click** move resize

- Star Menu
 - About

- Dialog



BU Resize

- **Alt click** move resize

keyboard Too

- 3 states: Normal / Full / Mini

Double click

Double click in top 10%

- keyboard

Space

Normal ↔ Full

Ctrl Space

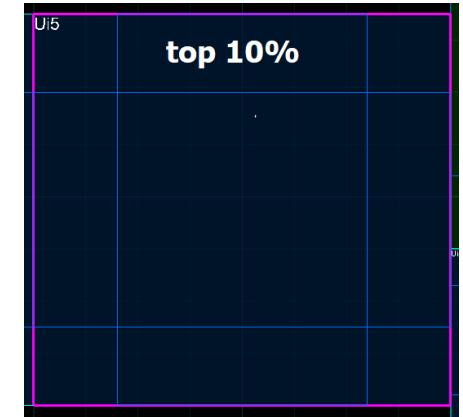
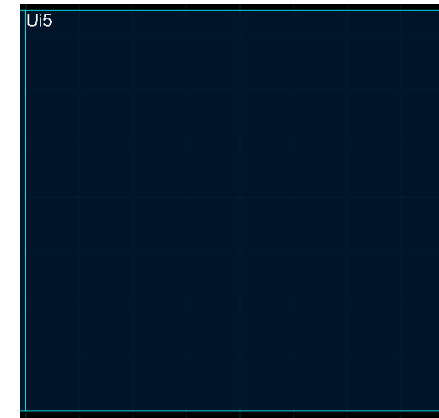
Full

Ctrl Arrows

Move

Ctrl + | -

Size

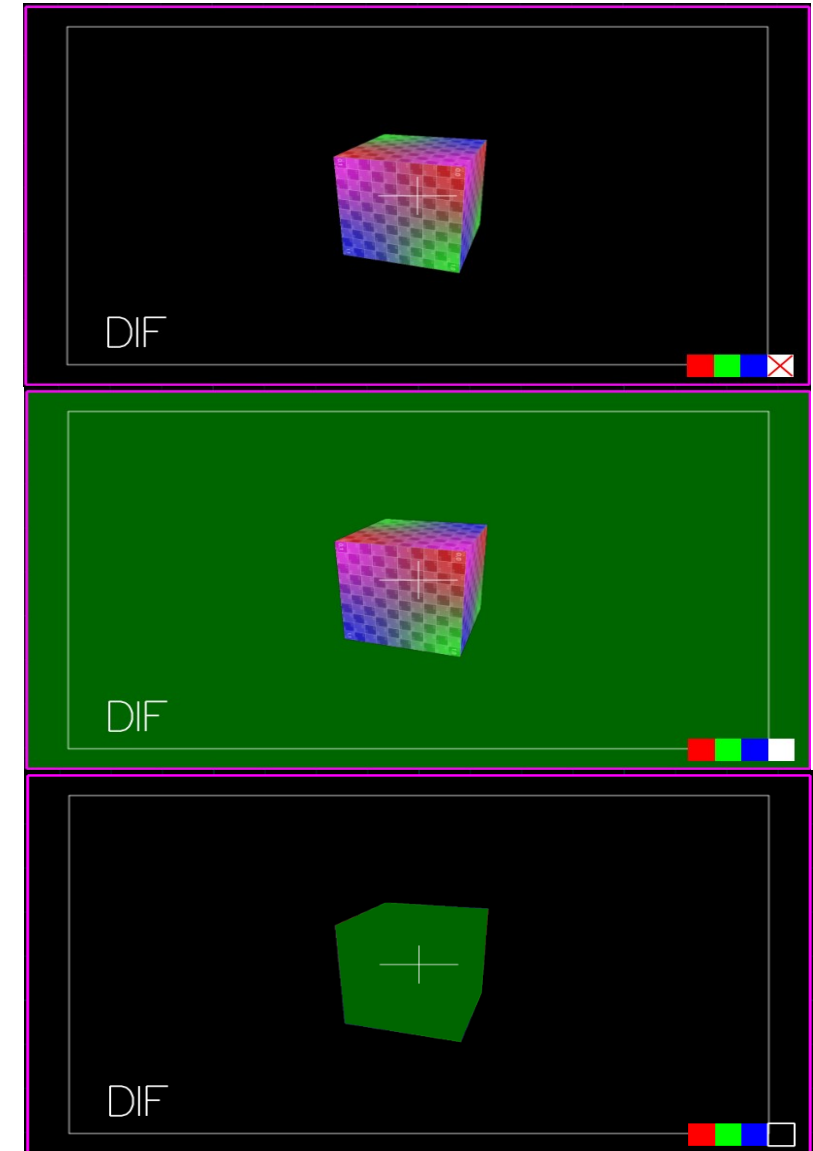
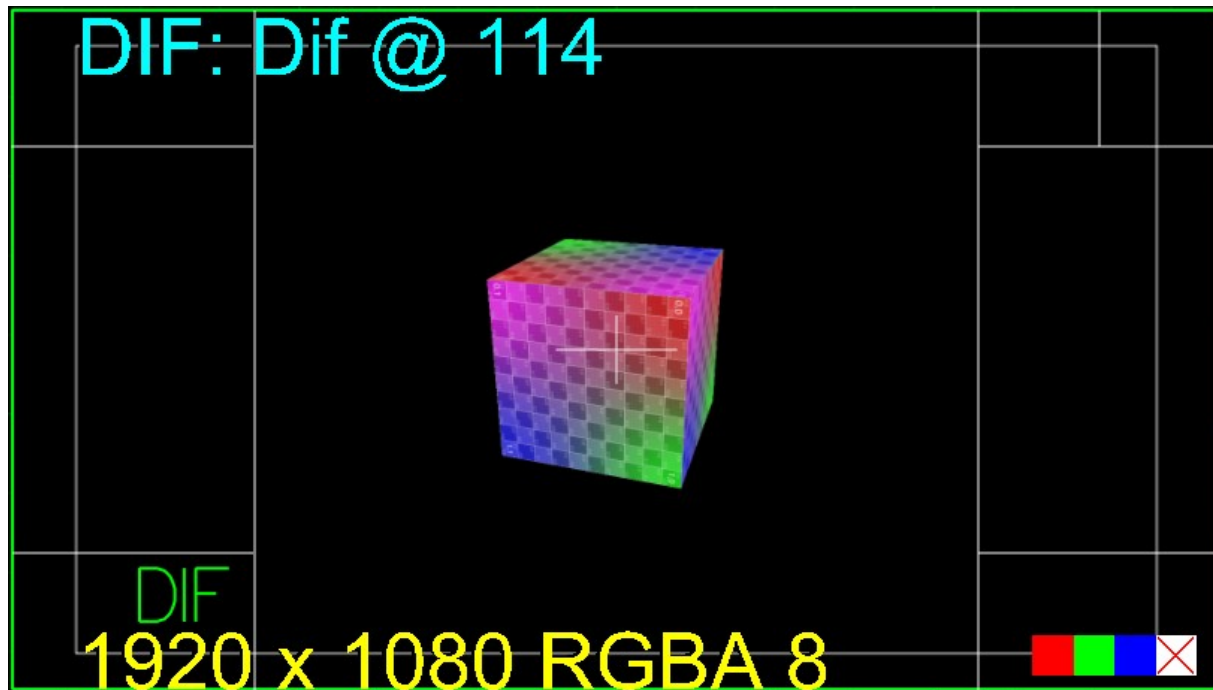


BU_MONITOR quick introduction

- Click zones

Click Right Middle

change how alpha is displayed



Flatland

- Original interface

where the C++ can be seen and used
a whole world we will explore later

F10 → Prefs

- Menus

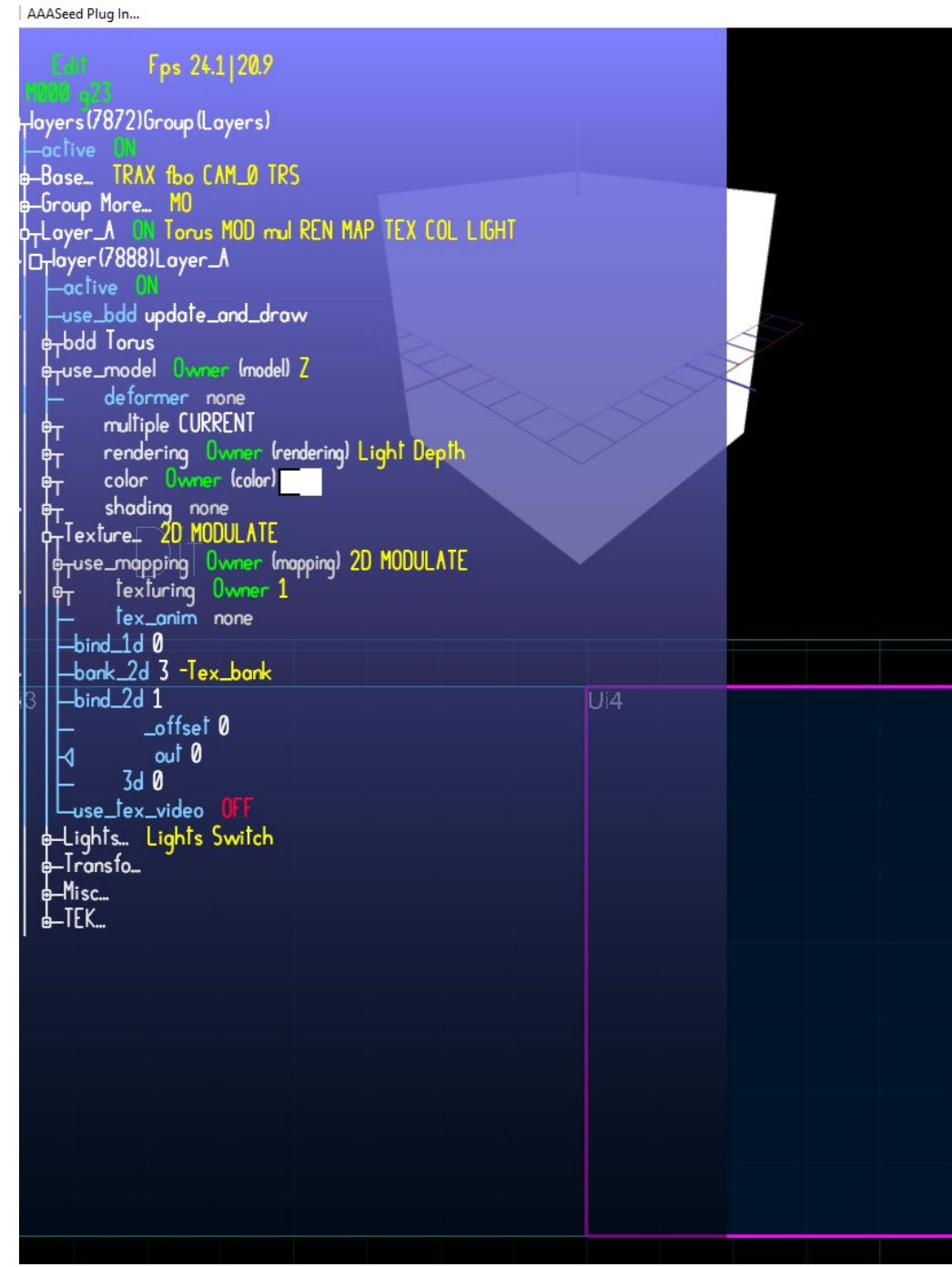
param / main

- **GaBuZoMeu**

top of main menu

- Precedence on event / keyboard

Live it off for now → **Tab**

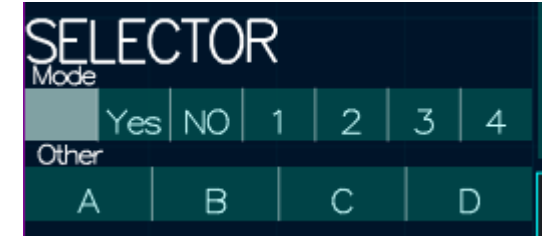
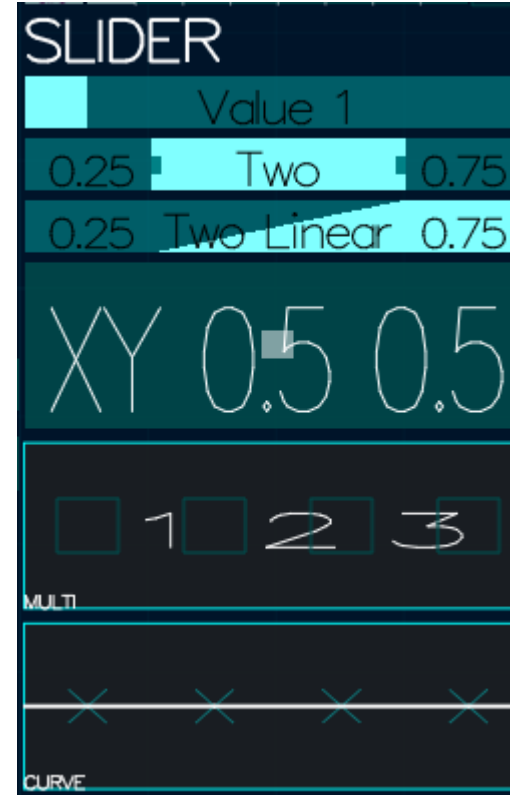


Basic BU: for bottom to top

- Click on Train_BU icon



- **BUTTON**
- **SLIDER**
- **SELECTOR**
- **BU_TEXT**
- current BU



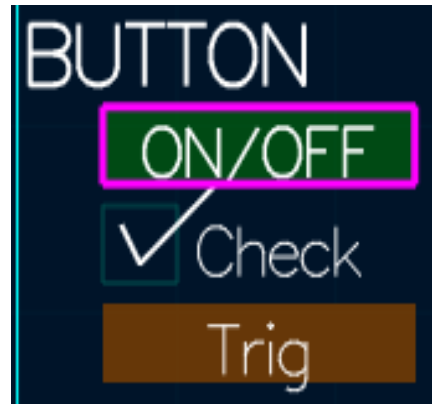
get keyboard
violet rectangle
move with arrows

BUTTON

- On/Off

2 versions

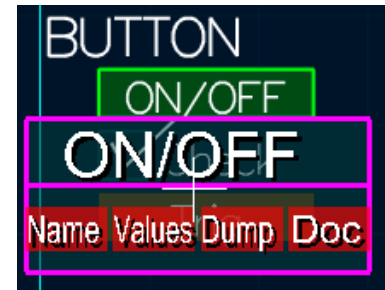
green / Red
checkbox



Trig

- **BUTTON** Multiple
- **BUTTON** Menu
- Access to Star Menu

Go out quick



- Confirmation **BUTTON** (e.g. debug)
- Could be moveable too



SLIDER

- Slide

Go up

then angle

Shift lock on values

Ctrl Alt → sensibility

- Min / Max

- Edit

Double Click or Dialog

- keys

+ | - | * | /

→ change

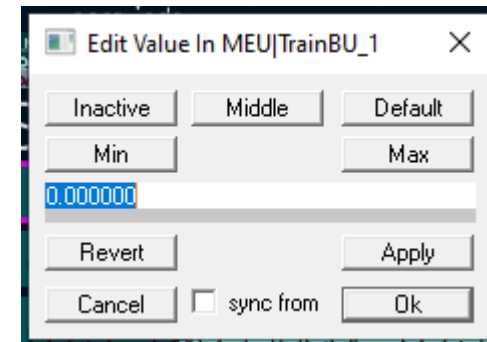
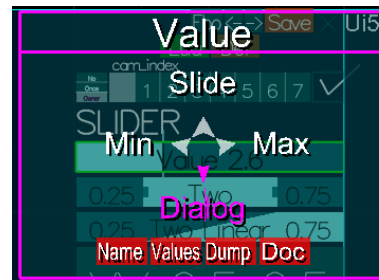
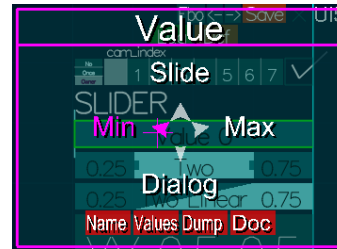
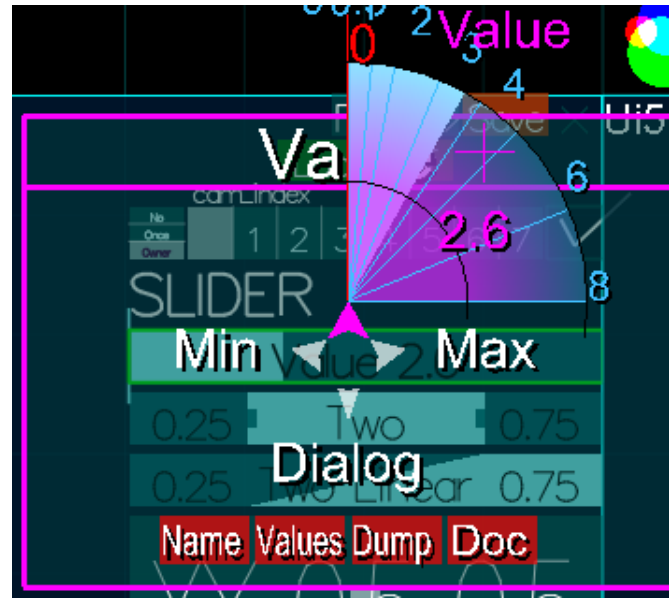
Enter

→ inverse

.

→ floor integer

- Floating point or integer



Back to BUTTON

- Slide not for now (2024 Oct) but will come

Go up

then angle

Shift lock on values

Ctrl Alt → sensibility

- Min / Max not for now (2024 Oct) but will come
- Edit not for now (2024 Oct) but will come

Double Click or Dialog

- keys

+ | - | * | / → change

Enter → inverse

. → floor integer

- boolean or integer

Other **SLIDERs**

- **SLIDER_TWO**

Range / Linear

depend where you click first

- **SLIDER_XY**

2 values

- **SLIDER_MULTI**

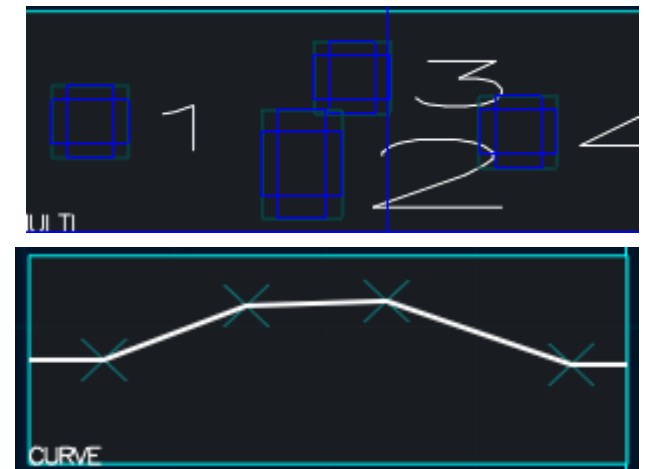
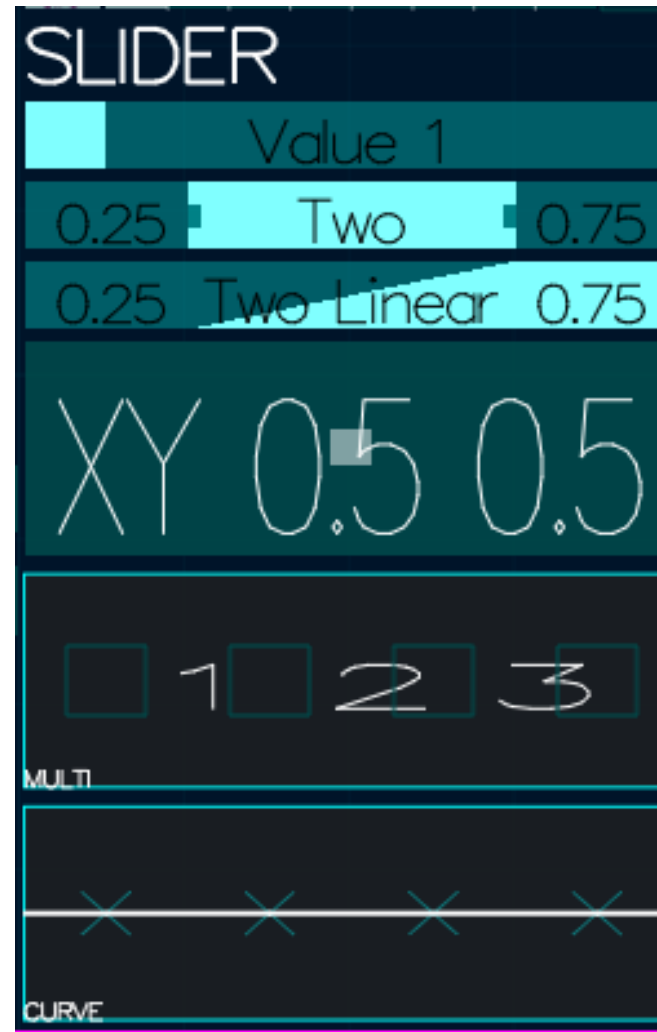
x/y and size_x/y

- **SLIDER_CURVE**

linear curve for now

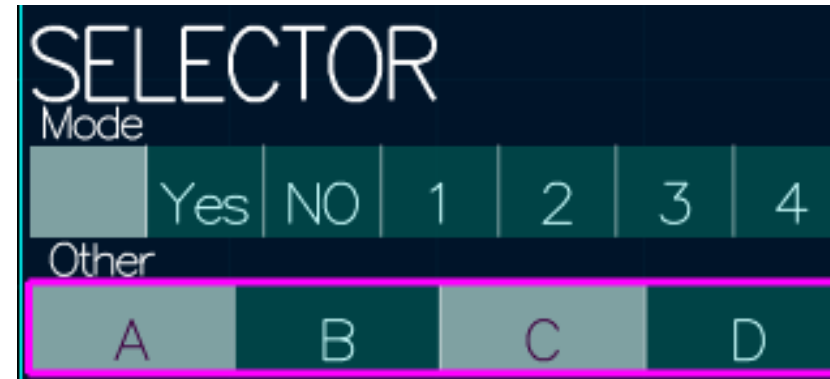
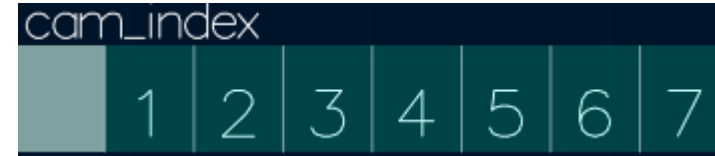
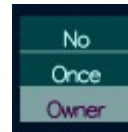
insert → Add point

delete → remove point



SELECTOR

- Access to Star Menu
like **BUTTON**: go out quick
- Keep mouse down
Change with mouse
- Use keyboard as for a **SLIDER**
+ and - mainly
- Multiple selection an option



BU_TEXT

- Editable or not

Double Click

- **BU_TEXT INFO**

Yellow in general

display info

- Access to Star Menu

like **BUTTON** and **SELECTOR**: go out quick

- Will Evolve
- Dialog Will Evolve
- Virtual keyboard exist not functional at the moment (2024 Sep)

Composite BU

- **BU multiple**



use a hidden SELECTOR

- **BU menu**

use a selector in an optional window

move to keep it open

close BU at top right



Some BU_VIZ widget/wizard

• BU_ALIVE show/control rendering F3

• BU_FPS Frame Per Second



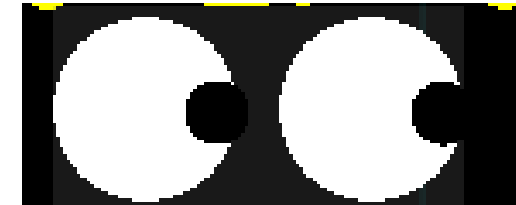
FPS 153 / 145

Flatland Too

• BU_TIME show time

23 : 26 58

• BU_EYE show mouse direction



• BU_CAM control edition of camera

CAM Locked

• BU_SEND control synchronisation between machines

NO SEND

• BU_MEM show memory used

1165

problem if it increase continuously

• BU_POWER show power and plug status

Power : Plugged 92%

• BU_BLOB show number of contact

-1-

MU Module Unit

MEU Module Editable Unit

- MEU base to encapsulate and manipulate functionality
- MU compact part of the MEU

Icon  + Slider 

- slider value is the alpha of the MU / MEU

Click on slider → StarMenu Slider but also MU

Click on icon → move

Alt Click → move/resize

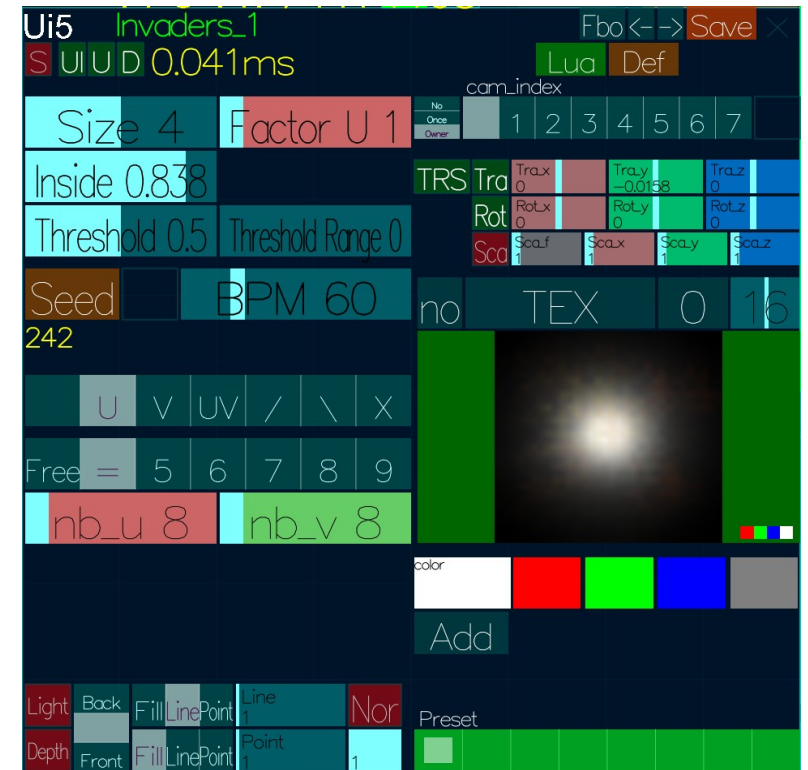
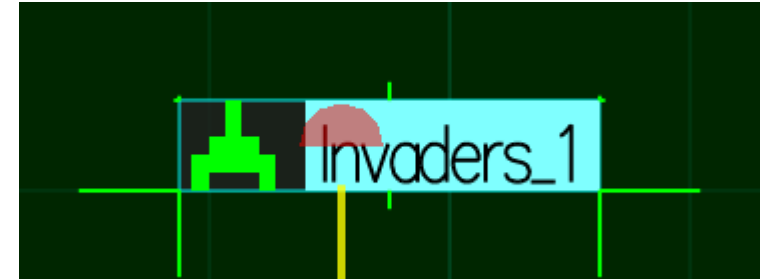
- State and Position of the MU control the rendering order

Slider value > 0 → On

vertical first: **bottom to top**

then horizontal: **left to right**

orange line show the **rendering chain** (more later)



MEU and UIx

- Click on **MU** icon → **MEU** in UI5
 - UI5 by default
 - UI6 Folder/Directory
- Links
- **MU** StarMenu to choose UI
- **MEU** StarMenu on background
- Close **BU** at top right
- Red Background ↔ not rendered
- Uix are infact **BU** too: **BU_MEU**
- Drawing change with distance
speed optimisation



MEU Preset

- At the bottom right of the **MEU**
- Load/Save the state of a **MEU**

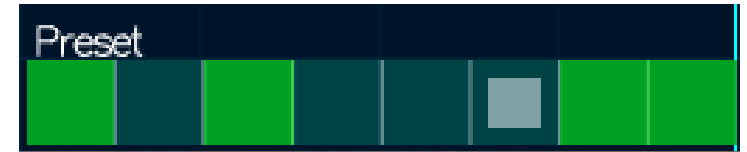
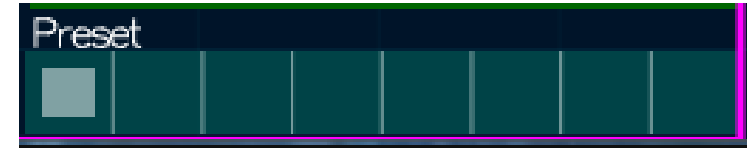
in fact load/save the values of included **BUs**

not always all, at the author discretion

- preset exit ↔ green color
- **Click** **load**
- **Ctrl Click** **save**
- **Ctrl Alt Click** **delete**
- Fixed number by MEU Type

defined in the code for now (2024 Sep)

e.g. meu:get_preset_nb() return 24 end



Preset							
			4			8	12
		16			20		24

More on MEU

- MEU is a module of fonctionnality, in fact in computer term it is an object
 - it usually render but can be otherwise
 - receive/send data
 - analayse image
 - control a device (e.g. plotter, projector, Dmx, Arduino...)
 - ...
- 2 main methods (function) are called every frame by a render() method
 - update() prepare so draw() will be as fast as possible
 - draw()
- 1 method is called when the MEU interface is visible
 - update_ui()

MEU real nature

- It is a **Lua object**

Name is `MeuType_InstanceName`

- It live in a **directory**

- Everything is readable










and can be edited

less and less needed

but some like it

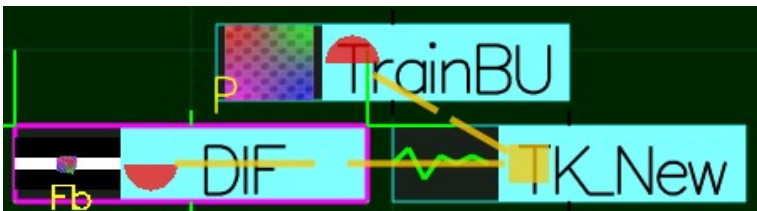
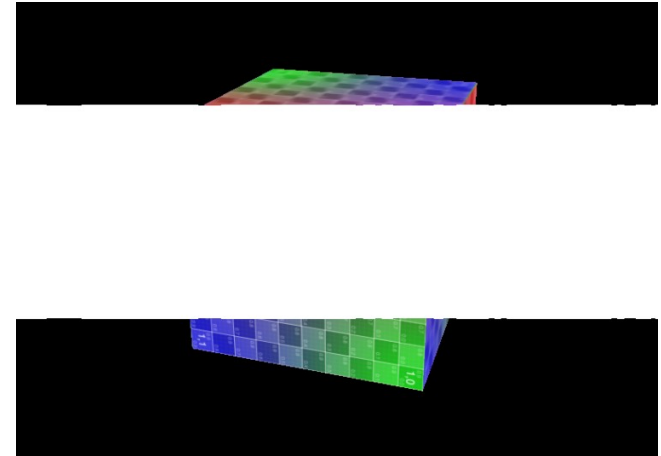
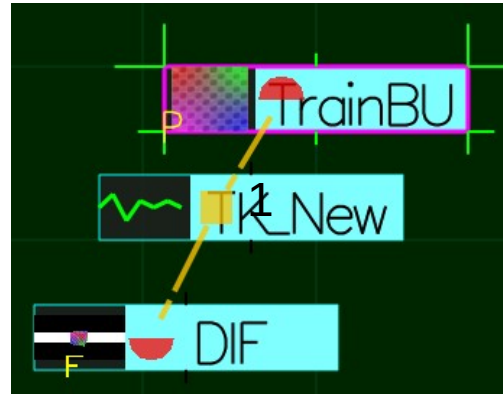
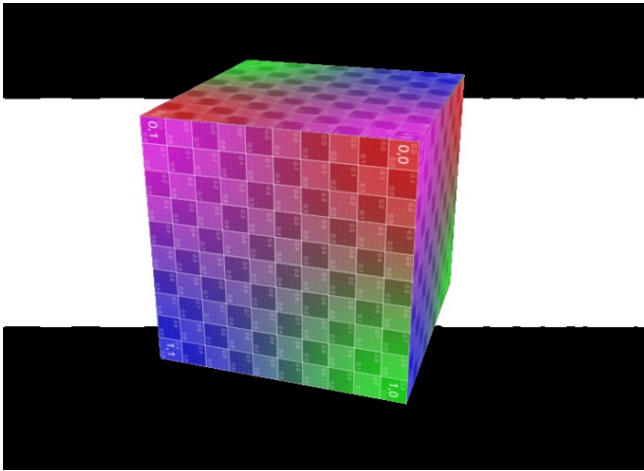
MEU Bar



- Title (top left in white or green if current) made of : MEU_name (short name)
MEU_name made of type (left) and instance (right): Type_Instance
name and short name between ()
- S UI U D **BUTTON**s (left bottom) 
switch to activate/deactivate **Send update_UI** Update and **Draw**
- Execution time (next right) in millisecond 
- A **SELECTOR** (middle) called tab to switch between different block of interface 
- Save (right) **BUTTON** which save only the **MEU** state 
- X or Close **BUTTON** (top right) close the **MEU** UI 
- Lua **BUTTON** open the lua script that the MEU use in an associated text editor 
- Def **BUTTON** trigger a redefinition of the **MEU** interface 
- Fbo **BUTTON** open the used **Fbo** (more soon) in the previous **BU_MEU** 
- Arrow **BUTTON** let you navigate in the rendering chain (more soon) 

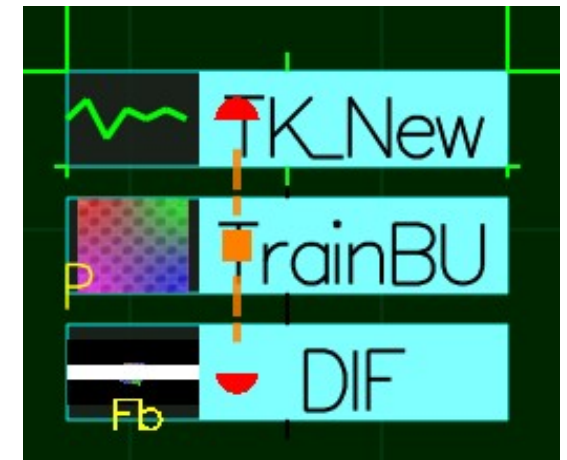
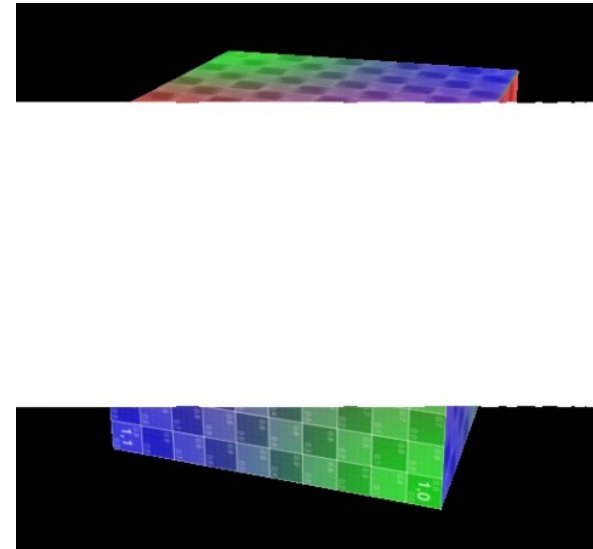
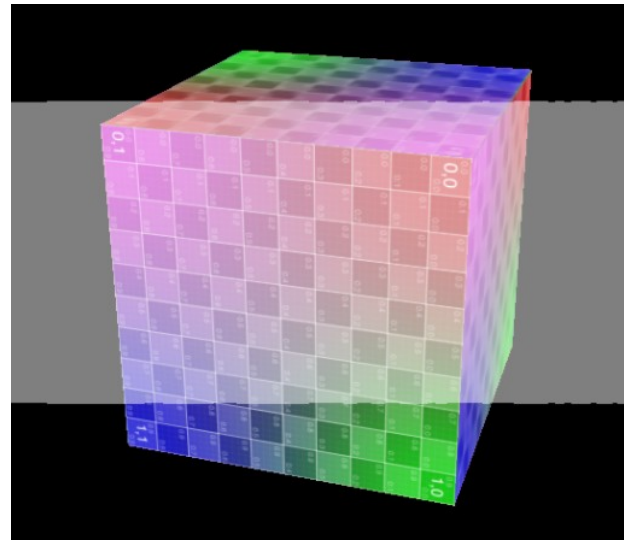
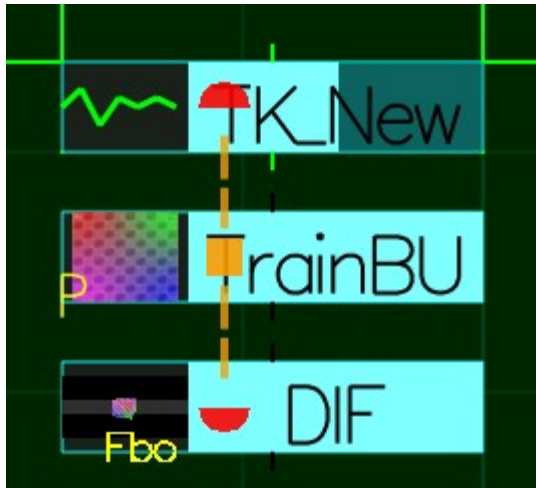
Rendering chain

- State and **Position** of the *MUs* control the execution order
 rendering order when MEU do rendering
- **bottom to top**, then **left to right**
- orange line **not a cable** just a **visualisation** of the rendering chain

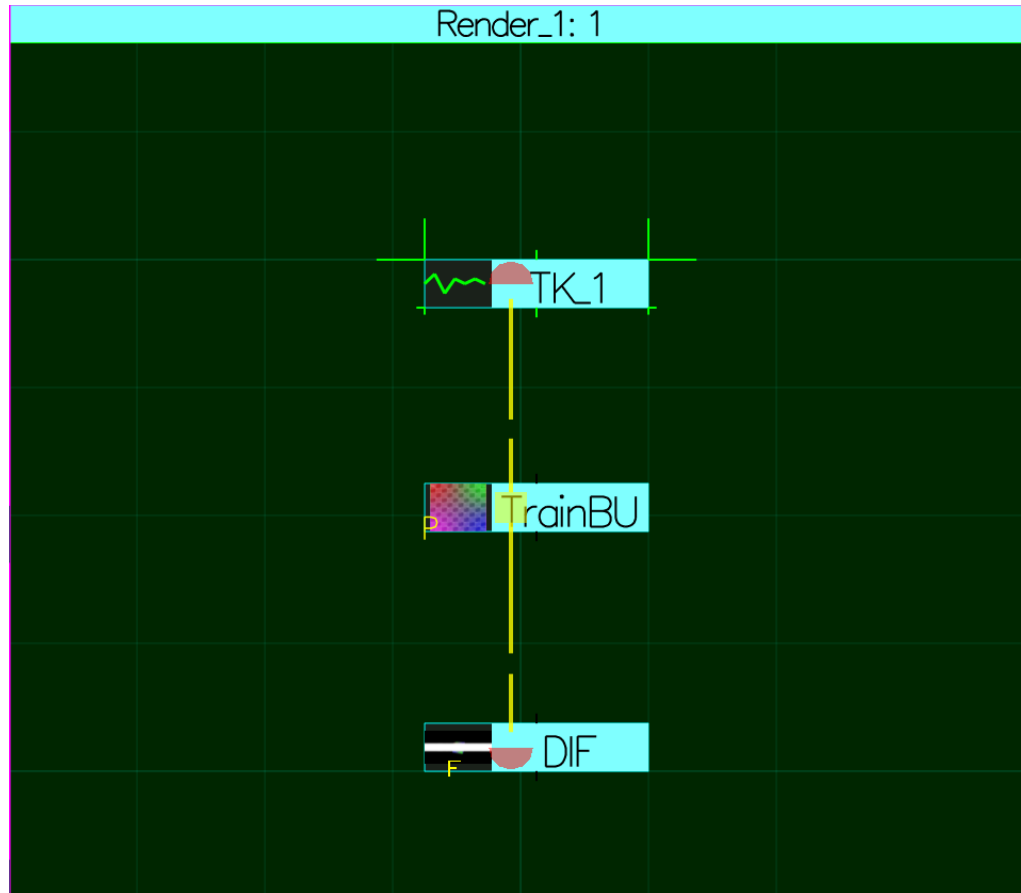


Alpha Opacity Transparency

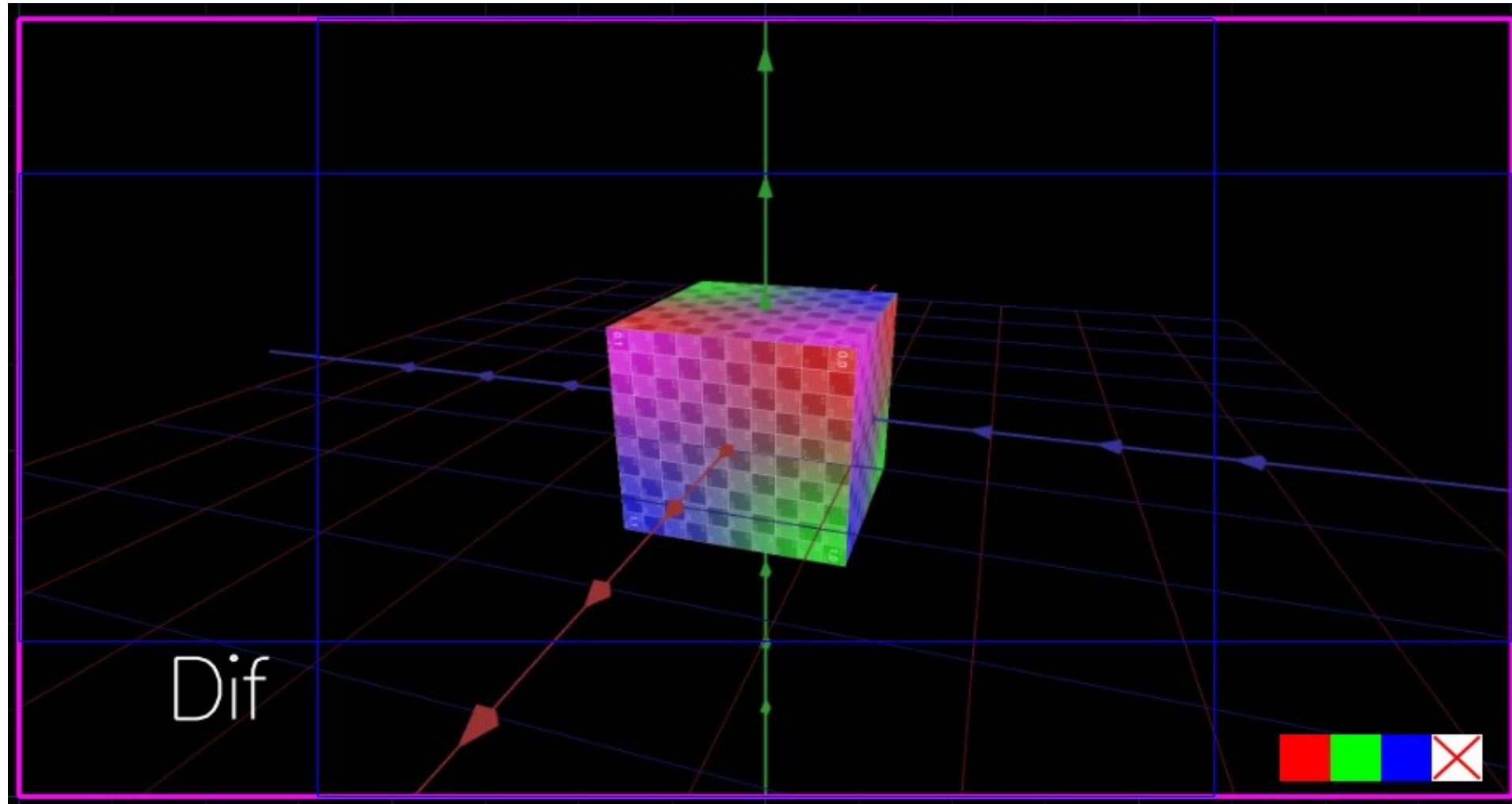
- **MEU/MU** slider value more than 0 → **On**
- when possible this value is the alpha level of the **MEU** (Opacity)



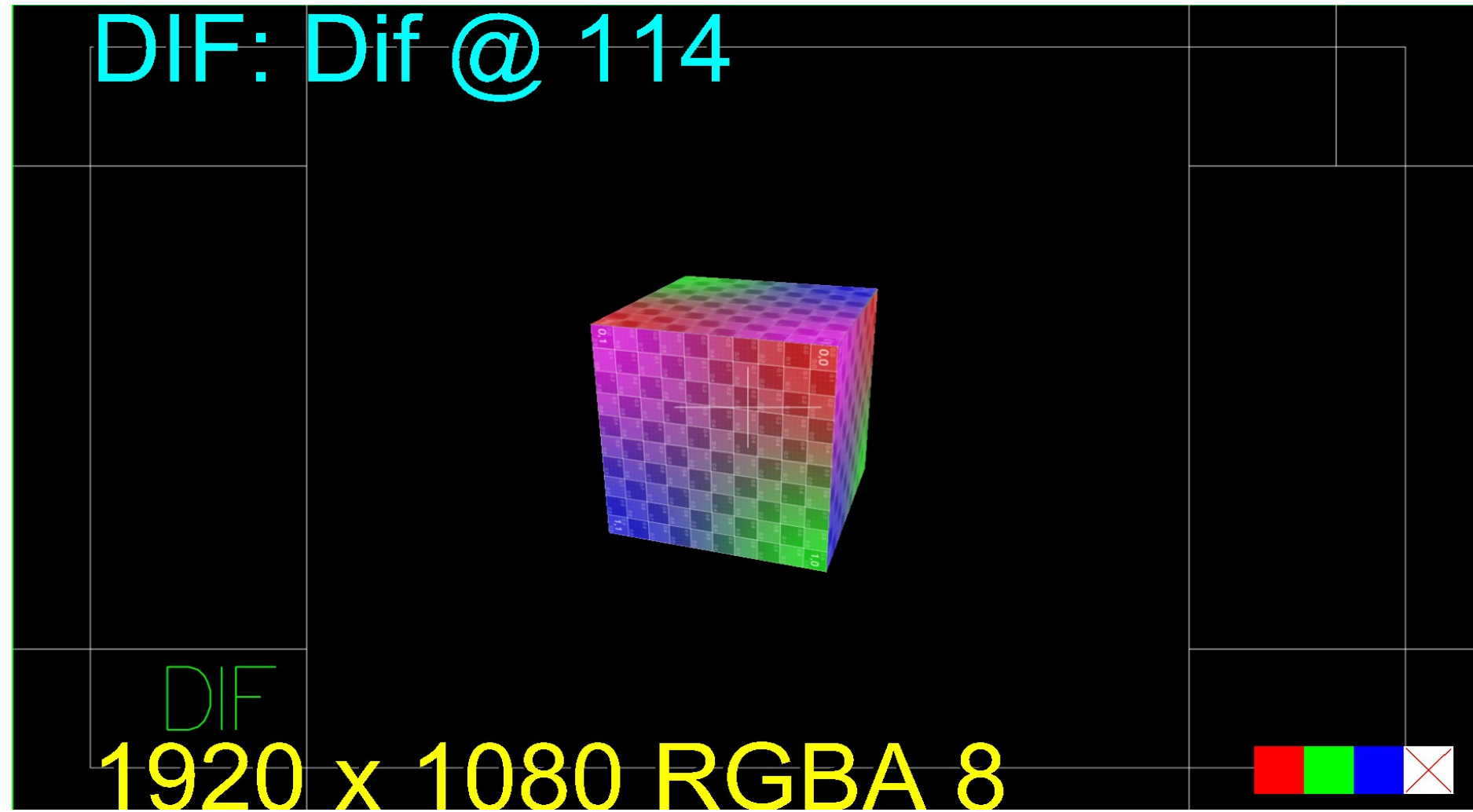
Rendering Chain BU_RECT



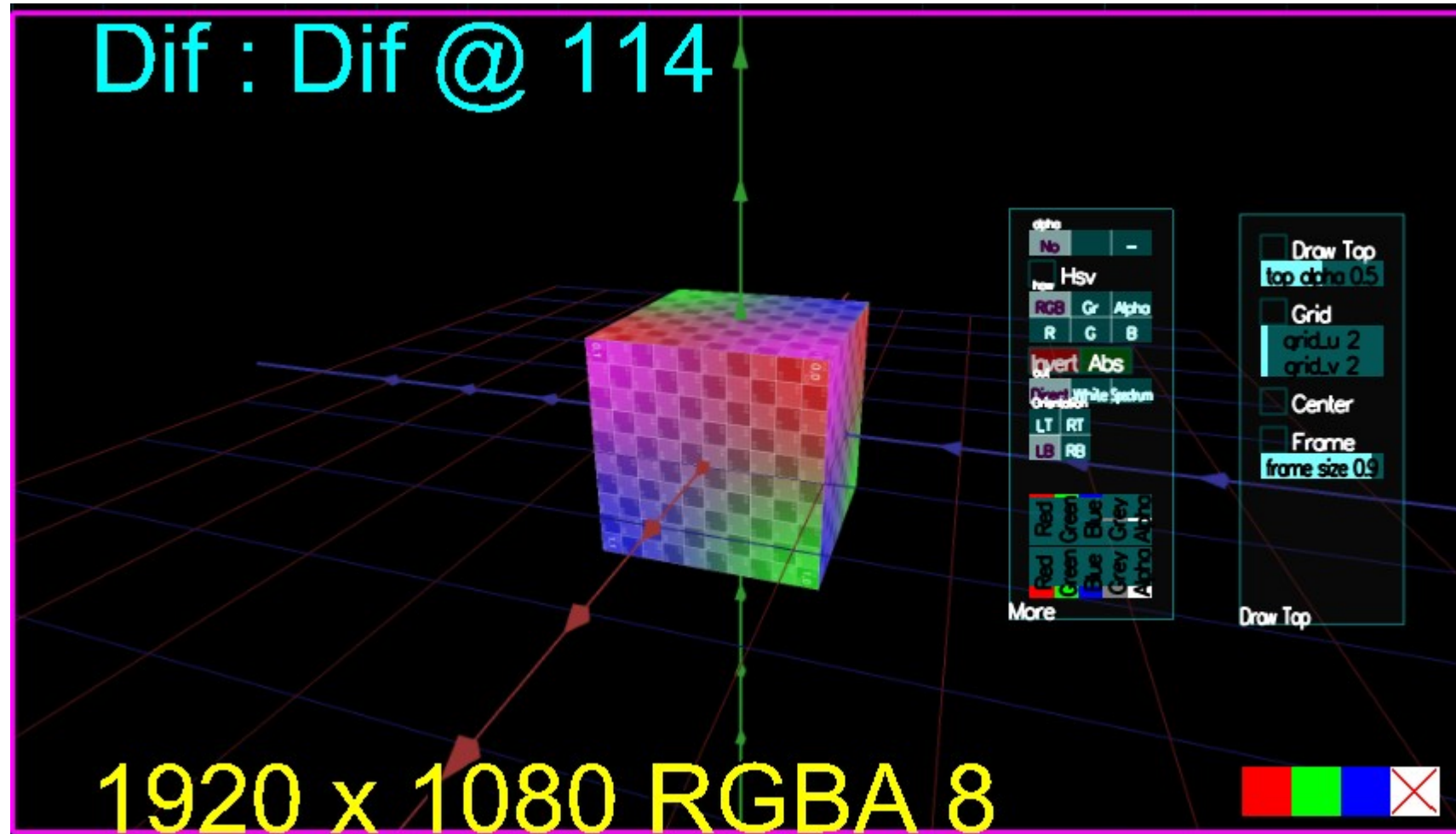
BU_MONITOR can be moved with Alt



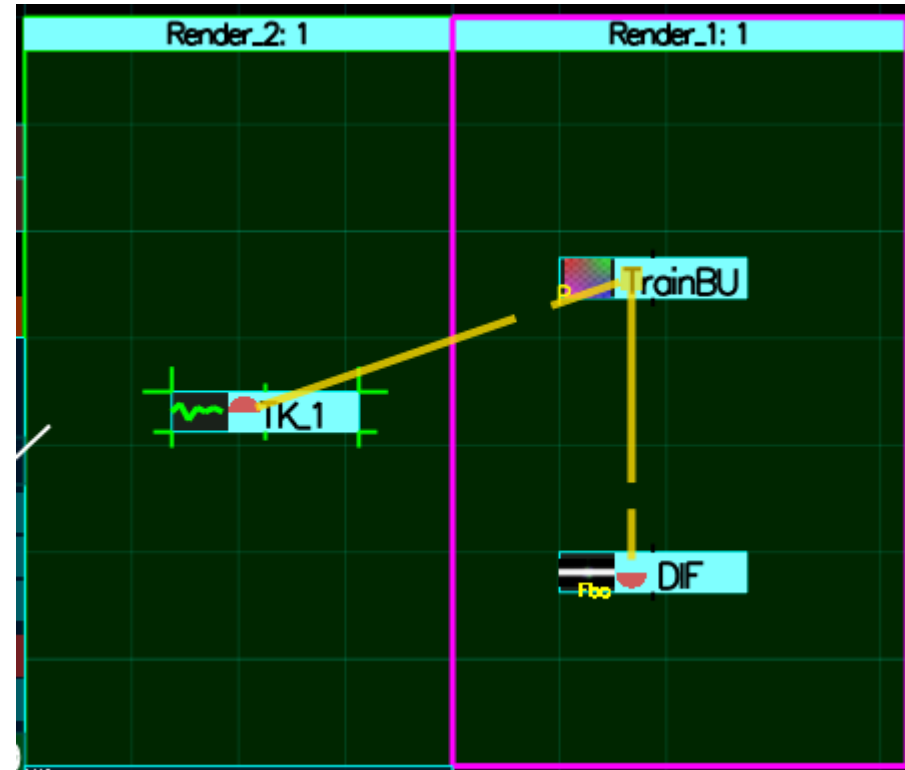
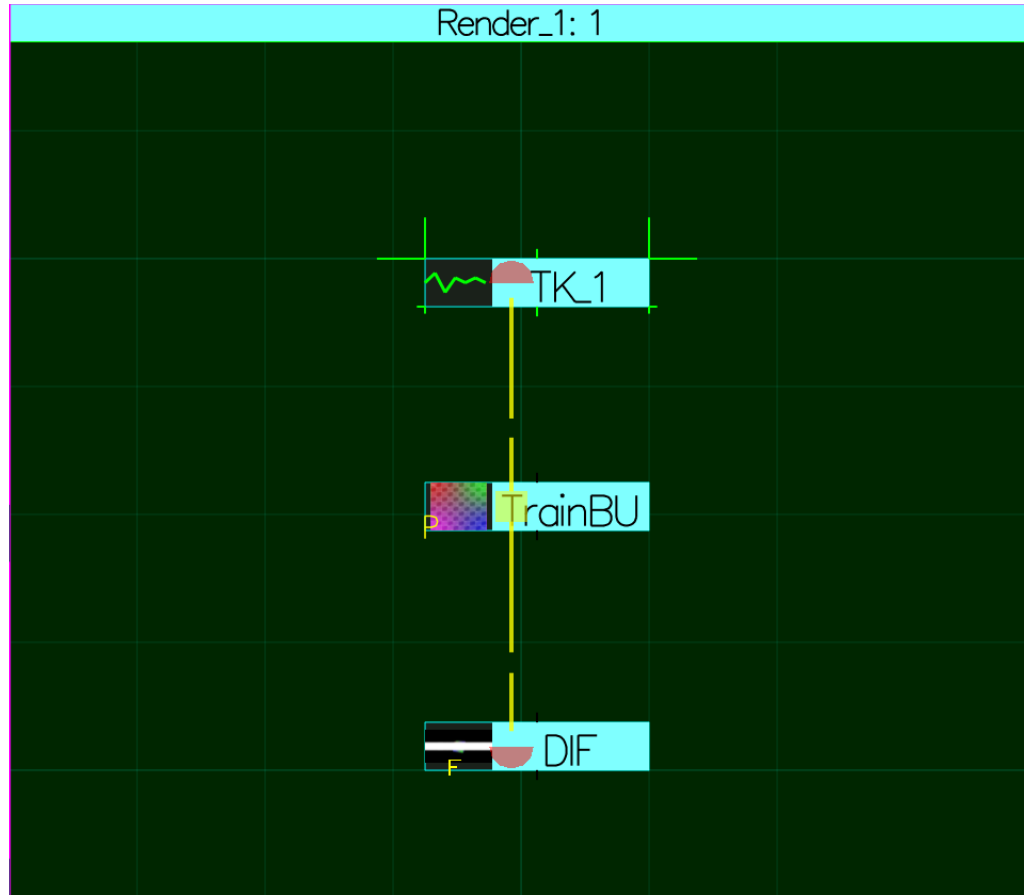
BU_MONITOR Click zones



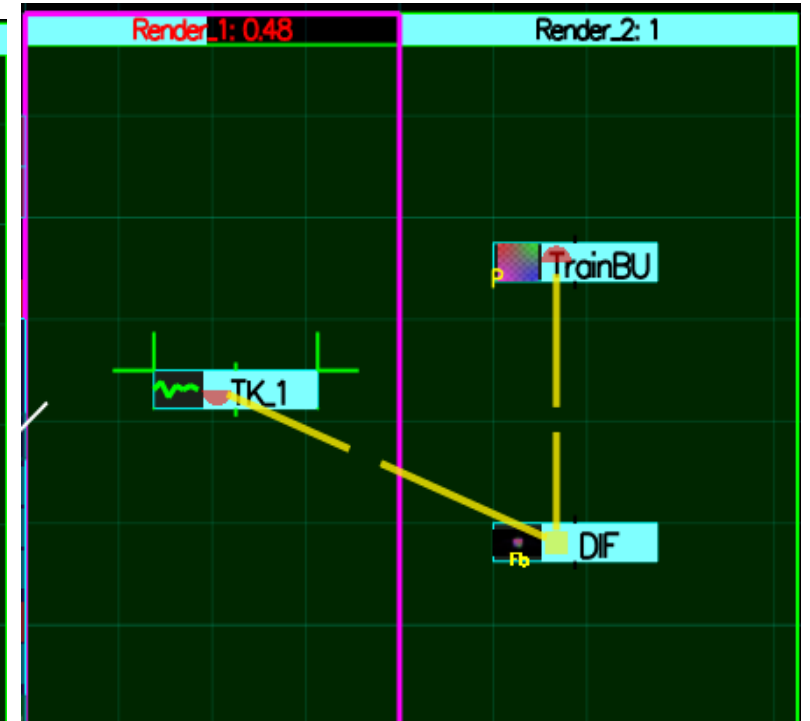
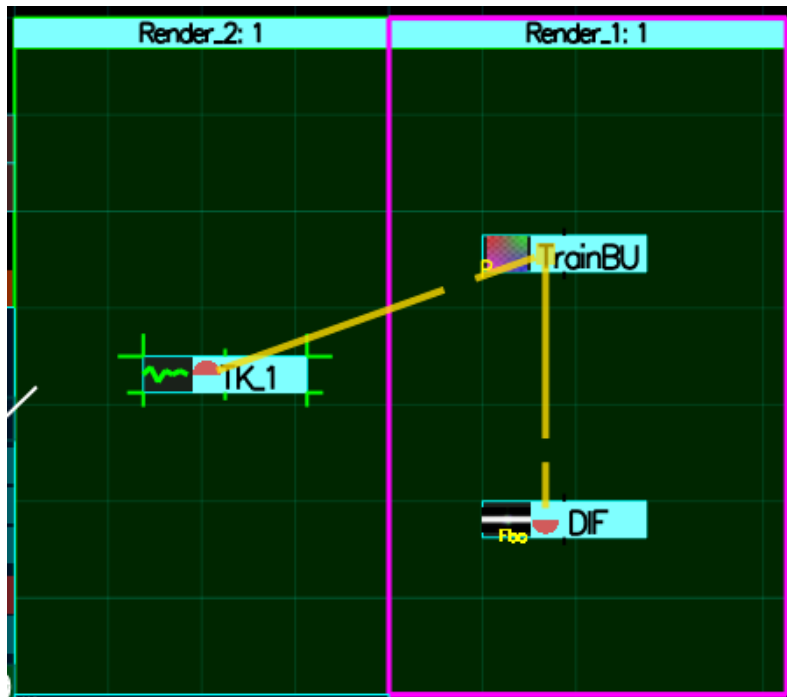
BU_MONITOR



Rendering Chain BU_RECT

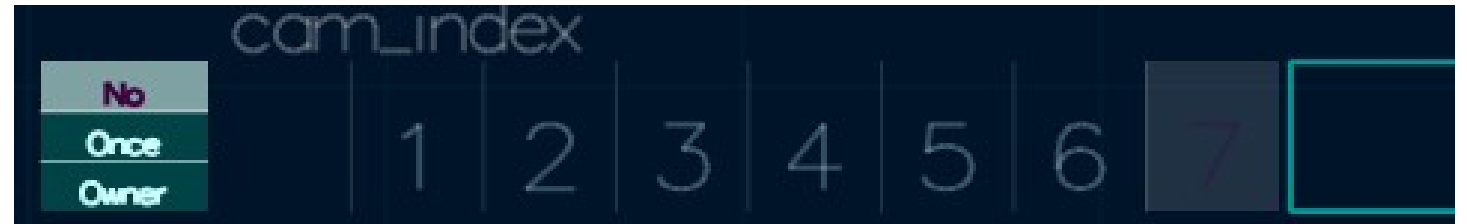
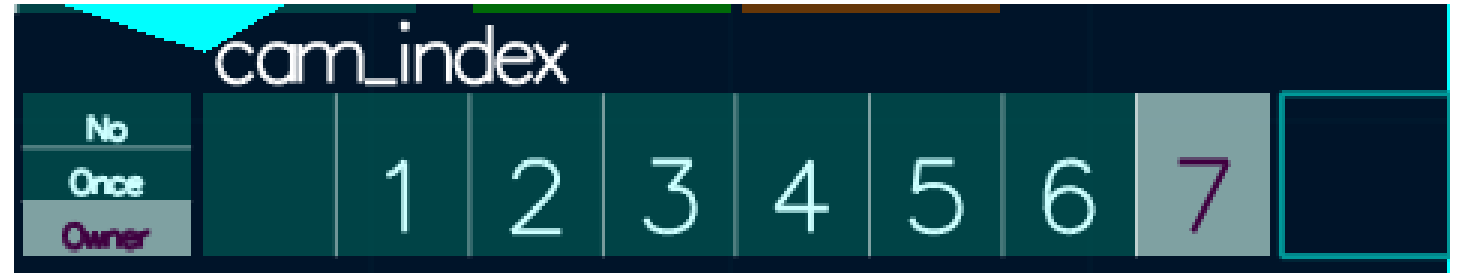


Rendering Chain BU_RECT



Cameras

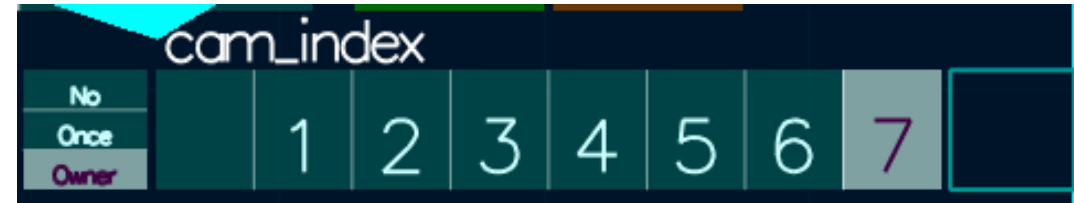
- No / Once / Cur
- Camera selector
- Button axe show



StarMenu → name

- | | |
|------------------------|------------------------------------|
| • Click | Read current Camera |
| • Ctrl Click | Write current Camera |
| • Ctrl C/Ctrl V | Copy / Paste current Camera |

Camera Edit



- Middle Click to Start
 - Wheel to zoom/dolly
 - Mouse to rotate
 - Middle Click Drag to move
- Left Click to accept
- Right Click to go back to start position

BU_MESS

```
# LUA : BU_CAM | viz_CAM : ----- GABU_OBJ unused key 9
# LUA : BU_CAM | viz_CAM : did not used key 9
# LUA : GARDEN | garden : try to use key 9
# LUA : GARDEN | garden : do_key( key=9 )
# LUA : do_key( self=GARDEN | garden, key=9 )
# LUA : GARDEN | garden : ----- GABU_OBJ unused key 9
# LUA : GARDEN | garden : did not used key 9
# LUA : GABU_SINGLETON : KEY NOT USED 9
# LUA : GABU.do_key_custom_def() key 9 Unused
```

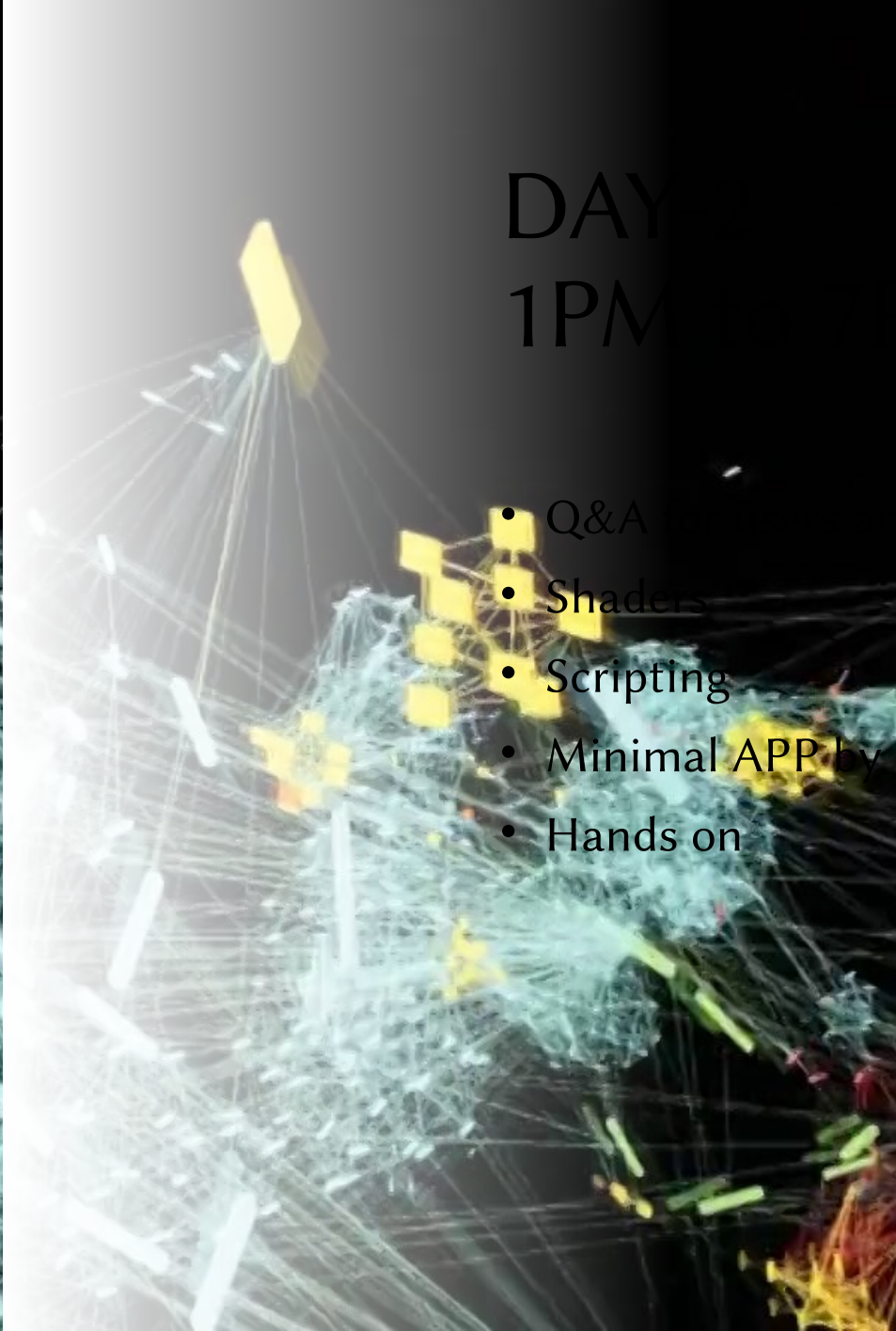
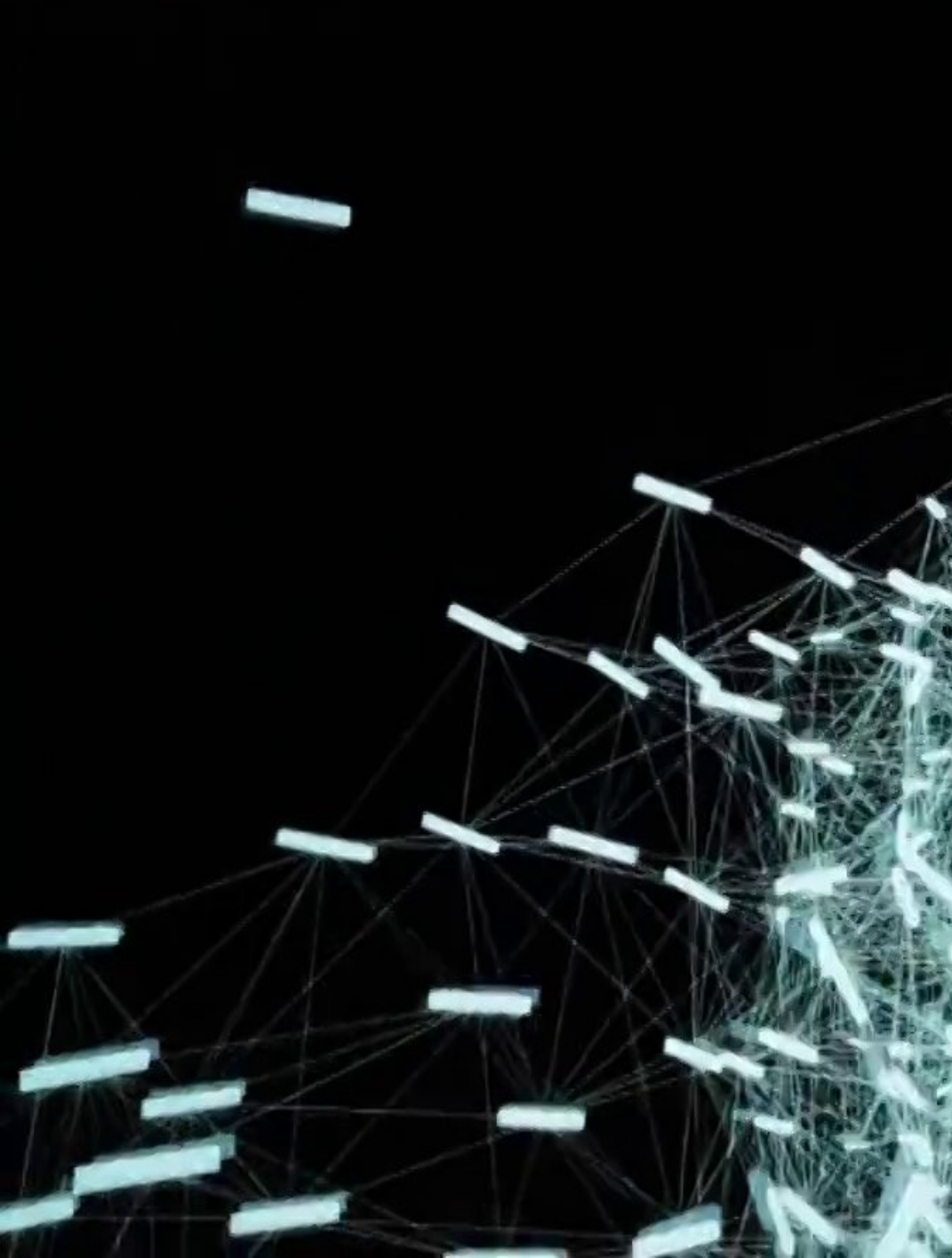
Mess

Scroller

Try Double Click

BU_SHOW

```
Multi blob_nb->0  
BUSSof_GA_top_top->0/0  
BUSSof_GA_top->220/0  
BUSSof_GA_regular->7/7
```



DAY 1

1PM - 3PM

- Q&A with experienced artists
- Shaders
- Scripting
- Minimal APP by trainee
- Hands on

File system Raw Approach

- AAADoc

 - lua_aaaseed_draw.lua

 - lua_aaaseed_interface.lua

- AAACore

 - where developer working on the core works

- Visual code workspace

- AAAUser

 - Duplicate folder

 - Rename it

 - Rename User

- APP

 - Create a New App just for you

 - Duplicate an existing one

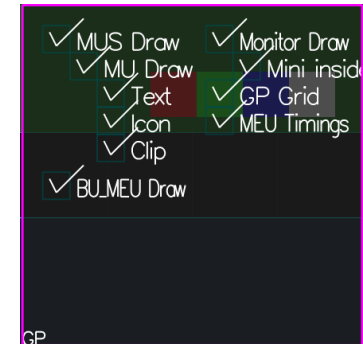
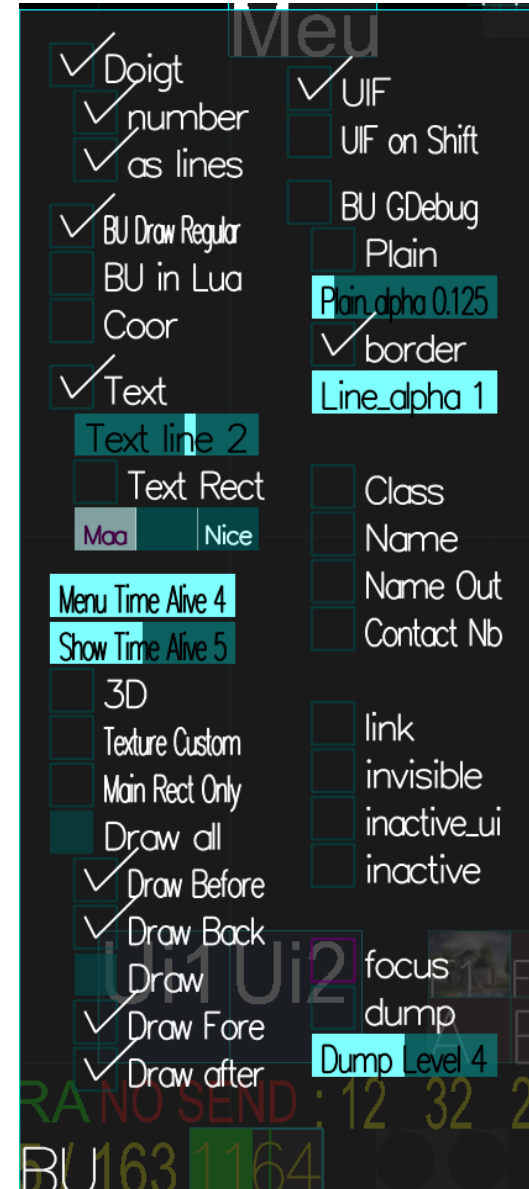
 - Quick navigation in it

- Fix

Window GA BU GP

- Garden Party
- Ctrl S, Esc/Esc Save it

GABUGP



Doc Window

- Documentation
- Lua inspector

The screenshot displays the 'Doc Window' interface, which is divided into several sections. On the left, there is a sidebar with a 'Show' button and three tabs: 'Private', 'All', and 'Public'. Below these tabs are two buttons: 'Global' and 'GaBu Classes'. The main area of the window is a grid of class names, each highlighted in green. The classes are arranged in four columns and ten rows. The first column contains: AAACAM, Balue, BLOB, BU_ALIVE, BU_CREATE_MEU, BU_KEY, BU_MENU, BU_OBJ, BU_SEND, BU_TEXTURE, BU_WINDOW_LIST, BUS, CALAGE, and COLOR_REF. The second column contains: APP, BDD_CLEAR_SCREEN, BLOBS, BU_BLOB, BU_DOC, BU_LIST, BU_PB, BU_SHADING, BU_VIZ, BU_WWW, BUS_CTX, CELT, and CREATURE. The third column contains: APP_FACTORY, BDD_FBX, BOID, BU_CAM, BU_EYE, BU_MAAEB, BU_MEU, BU_POWER, BU_SHOW, BU_WATCH, BUI, BUSS, CHANGER, and DATAGRID. The fourth column contains: APP_GP, BIND_TEX, BU, BU_COLOR, BU_FPS, BU_MEMORY, BU_MONITOR, BU_RECT, BU_TEXT, BU_WINDOW, BUP, BUTTON, CHANGERS, and EVENT. At the bottom of the window, there is a 'Doc' label and a 'Use' button. To the right of the 'Use' button is an 'Add' button and a large 'BU' label. On the far right, there is a 'Scroller' button and a small 'color' label. The top right corner of the window shows a Lua script snippet with comments and code.

Info

Show
Private All Public

Global
GaBu Classes

Ui4

Doc Use Add BU

Scroller

LUA: BU_DOC(Doc: BUFlip_full_page_and_mini() y: 0.16473110984788
LUA: Mouse grabbed
LUA: BU_DOC(Doc: will be current bu now
LUA: Mouse grabbed
LUA: BU_DOC(Doc: BUFlip_full_page_and_mini() y: 0.16492750841935

AAACAM	APP	APP_FACTORY	APP_GP
Balue	BDD_CLEAR_SCREEN	BDD_FBX	BIND_TEX
BLOB	BLOBS	BOID	BU
BU_ALIVE	BU_BLOB	BU_CAM	BU_COLOR
BU_CREATE_MEU	BU_DOC	BU_EYE	BU_FPS
BU_KEY	BU_LIST	BU_MAAEB	BU_MEMORY
BU_MENU	BU_MESS	BU_MEU	BU_MONITOR
BU_OBJ	BU_PB	BU_POWER	BU_RECT
BU_SEND	BU_SHADING	BU_SHOW	BU_TEXT
BU_TEXTURE	BU_VIZ	BU_WATCH	BU_WINDOW
BU_WINDOW_LIST	BU_WWW	BUI	BUP
BUS	BUS_CTX	BUSS	BUTTON
CALAGE	CELT	CHANGER	CHANGERS
COLOR_REF	CREATURE	DATAGRID	EVENT

Lua example



- Example of drawing with lua using OpenGL directly
- The concept look like processing

iterative process

How to deal with errors

Flatland / Pref(F10)/ Master / lua / error_trig_editor



Meu Window

- Navigate MEU
 - Prototype
- Create New MEU
- Select by Tag
- Select by Name

The screenshot displays the 'Meu Window' interface. At the top, it shows 'MEU nb 103/121'. Below this, there's a list of tags on the left, including 'Self', 'All', 'Proto', 'No Tag', 'All', and 'Tag'. The main area shows a grid of MEUs with their names and tags. The tags are color-coded: red for 'Proto' and green for 'All'. The MEUs are arranged in a grid, with some having additional icons or symbols next to them. At the bottom, there's a 'Use' button and a 'BU' button. The interface also includes a 'Scroller' on the right side and a 'Meu' label at the bottom left.

Tag	MEU Name	Icon
Proto	2054	
Proto	Blur	
Proto	CaptureRect	×
Proto	ColorCurve	
Proto	Dir	CV
Proto	DistField	
Proto	FaceUV	
Proto	Flex	
Proto	GridSel	Hexagon
Proto	ImgSend	⬆
Proto	KinFlipper	
Proto	LightPassV1	
Proto	MeshStatic	
Proto	MuEnd	⬆
Proto	AAAUtils	
Proto	Boid	
Proto	Clear	
Proto	DepthPick	
Proto	Displace	
Proto	ExShaderGrid	
Proto	Fbx	
Proto	FlexVideo	
Proto	Hexa	Hexagon
Proto	Invaders	
Proto	KinMove	
Proto	Lights	
Proto	Mondrian	
Proto	NdcAddBlur	
Proto	App	
Proto	Bullet	
Proto	Clous	
Proto	Derviche	
Proto	DisplaceCV	
Proto	ExShaderInstance	
Proto	FbxMatte	
Proto	FP	
Proto	HexCraze	
Proto	Kinect	
Proto	KinMoveAuto	
Proto	Marseille	
Proto	Monitor	
Proto	NdcBloom	
Proto	BlobDetect	
Proto	Cam	
Proto	CIPool	
Proto	DigitalProjection	
Proto	DisplacePart	
Proto	FaceTrak	
Proto	FieldGene	
Proto	Grab	
Proto	imgAnal	
Proto	Kinect1	
Proto	Lidar	
Proto	Materials	
Proto	MuBegin	
Proto	NdcBranching	

Core MEU

- Fbo
- Video
- PIP
- Out

DAY-3

10 AM to 3PM

- Specific Q&A session
- Lunch all together
- Use case implementation

